

DAM
Desarrollo de Aplicaciones Multiplataforma
2º Curso

AD
Acceso a Datos

UD 4
Modelo Objeto Relacional (ORM)

IES BALMIS
Dpto Informática
Curso 2019-2020
Versión 1 (03/2019)

UD4 – Modelo Objeto Relacional (ORM)

ÍNDICE

8. Hibernate y relaciones “muchos a muchos”

9. Anotaciones e Hibernate

8. Hibernate y relaciones "muchos a muchos"

En nuestra base de datos **bibliotecah2** habíamos dado por sentado que un autor puede escribir varios libros, pero cada libro pertenece a un único autor.

Esto, aunque es el caso habitual, no es cierto siempre, dado que un libro puede tener varios autores. Si queremos plasmar este detalle, deberemos crear tres tablas en la que podría ser una nueva base de datos denominada **bibliotecah3**:

```
CREATE TABLE autores (
  cod VARCHAR(5) PRIMARY KEY,
  nombre VARCHAR(60)
);

CREATE TABLE libros (
  id INTEGER PRIMARY KEY,
  titulo VARCHAR(60)
);

CREATE TABLE escribir (
  codautor VARCHAR(5),
  codlibro INTEGER,
  PRIMARY KEY (codautor, codlibro),

  FOREIGN KEY (codautor) REFERENCES autores(cod),
  FOREIGN KEY (codlibro) REFERENCES libros(id)
);
```

Si la relación se plasma de forma correcta (con una clave primaria compuesta, formada por las dos claves de las otras tablas, y con claves ajenas que hacen referencia a ambas), Hibernate creará solo dos clases:

- una para autores (y cada autor contendrá un "Set" de libros)
- y otra para libros (y en cada libro existirá un "Set" de autores).

El manejo en este caso será similar al que ya hemos visto, con la diferencia de que al anotar que un libro ha sido escrito por un autor, deberemos añadirlo dentro del "Set" de libros del autor o bien dentro del "Set" de autores de un libro, pero no en ambos sitios (lo que se añade usando una clase, será accesible desde la otra).

```

INSERT INTO autores (cod, nombre) VALUES
('WSHAK', 'William Shakespeare'),
('MCERV', 'Miguel de Cervantes'),
('FROJA', 'Fernando de Rojas');

INSERT INTO libros (id, titulo) VALUES
(1, 'Macbeth'),
(2, 'La Celestina (Tragicomedia de Calisto y Melibea)'),
(3, 'Don Quijote de la Mancha'),
(4, 'La tempestad'),
(5, 'La Galatea'),
(6, 'Los trabajos de Persiles y Sigismunda'),
(7, 'Novelas ejemplares');

INSERT INTO escribir (codautor, codlibro) VALUES
('WSHAK', 1),
('FROJA', 2),
('MCERV', 3),
('WSHAK', 4),
('MCERV', 5),
('MCERV', 6),
('MCERV', 7);

INSERT INTO libros (id, titulo) VALUES
(8, 'La última respuesta'),
(9, 'La buena vida'),
(10, 'Pulsaciones');

INSERT INTO autores (cod, nombre) VALUES
('AROFI', 'Alex Rovira'),
('FMIRA', 'Francesc Miralles');

INSERT INTO escribir (codautor, codlibro) VALUES
('AROFI', 8),
('FMIRA', 8),
('AROFI', 9),
('FMIRA', 10);

```

En este ejemplo podemos comprobar que hay el libro 8 está escrito por dos autores:

```

SELECT libros.*, autores.*
FROM libros, escribir, autores
WHERE
    libros.id = escribir.codlibro AND
    escribir.codautor = autores.cod;

```

id	titulo	cod	nombre
8	La última respuesta	AROFI	Alex Rovira
9	La buena vida	AROFI	Alex Rovira
8	La última respuesta	FMIRA	Francesc Miralles
10	Pulsaciones	FMIRA	Francesc Miralles
2	La Celestina (Tragicomedia de Calisto y Melibea)	FROJA	Fernando de Rojas
3	Don Quijote de la Mancha	MCERV	Miguel de Cervantes
5	La Galatea	MCERV	Miguel de Cervantes
6	Los trabajos de Persiles y Sigismunda	MCERV	Miguel de Cervantes
7	Novelas ejemplares	MCERV	Miguel de Cervantes
1	Macbeth	WSHAK	William Shakespeare
4	La tempestad	WSHAK	William Shakespeare

Si realizamos una nuevo proyecto **Hibernate3** seleccionando las 3 tablas tendremos que hacer un pequeño cambio respecto a **Hibernate2** en el método **verLibros()** para mostrar todos sus autores:

```
public static void verLibros() {  
  
    System.out.println("Mostrando todos los libros y sus autores:");  
    System.out.println("-----");  
    Query consulta = sesion.createQuery("from Libros");  
  
    List resultados = consulta.list();  
    for(Object resultado : resultados) {  
  
        Libros libro = (Libros) resultado;  
        System.out.println(libro.getTitulo()+" (" +libro.getId()+ "). Autores: ");  
  
        for(Object aut : libro.getAutoreses())  
            System.out.println(" => " + ((Autores) aut).getNombre());  
  
        System.out.println();  
    }  
}
```